

<b>Started on</b>	Thursday, 30 May 2024, 12:35 PM
<b>State</b>	Finished
<b>Completed on</b>	Thursday, 30 May 2024, 3:05 PM
<b>Time taken</b>	2 hours 29 mins
<b>Grade</b>	<b>460.00</b> out of 600.00 (77%)

Question 1

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Diberikan kode [House.java](#) sebagai berikut dan [IHouseBuilder.java](#) sebagai berikut

Implementasikan HouseBuilder.java yang mengimplementasikan interface IHouseBuilder. Spesifikasi tambahan sebagai berikut.

- HouseBuilder mampu untuk membuat house baru sesuai dengan argumen yang diberikan.
- HouseBuilder perlu dapat menyimpan state dari argumen yang diberikan sebelumnya
- Jika tidak dispesifikan nilainya, maka digunakan default nilai dari tipe tersebut, String → null, int → 0, dll


Berikut merupakan contoh hasil dari HouseBuilder

```
```java
public class App {
    public static void main(String[] args) {
        IHouseBuilder builder = new HouseBuilder();
        House house = builder.withColor("Red")
            .withFloors(2)
            .withRooms(3)
            .withOwner("John Doe")
            .withAddress("Jl. Merdeka No. 1")
            .build();
        System.out.println(house);
        // House{color=Red,floors=2,rooms=3,owner=John Doe,address=Jl. Merdeka No. 1}

        House house2 = builder.withRooms(5).build();
        System.out.println(house2);
        // House{color=Red,floors=2,rooms=5,owner=John Doe,address=Jl. Merdeka No. 1}

        IHouseBuilder builder2 = new HouseBuilder();
        House house3 = builder2.withColor("Blue")
            .withOwner("Jane Doe")
            .withAddress("Jl. Merdeka No. 2")
            .build();
        System.out.println(house3);
        // House{color=Blue,floors=0,rooms=0,owner=Jane Doe,address=Jl. Merdeka No. 2}
    }
}
```
```

Java 8

 [HouseBuilder.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict  | Description        |
|----|-------|----------|--------------------|
| 1  | 20    | Accepted | 0.07 sec, 27.83 MB |

| No | Score | Verdict  | Description        |
|----|-------|----------|--------------------|
| 2  | 20    | Accepted | 0.08 sec, 28.18 MB |
| 3  | 20    | Accepted | 0.15 sec, 28.14 MB |
| 4  | 20    | Accepted | 0.09 sec, 29.13 MB |
| 5  | 20    | Accepted | 0.11 sec, 28.18 MB |

Question **2**

Correct

Mark 100.00 out of 100.00

|              |       |
|--------------|-------|
| Time limit   | 1 s   |
| Memory limit | 64 MB |

## Inventory of Generic Boxes

Level: Easy

Tuan Syahul ingin menerapkan konsep generic pada OOP. Dia ingin membuat sebuah kelas Box yang dapat menampung objek apapun dan terdapat sebuah inventori yang bisa menyimpan box-box tersebut. Implementasikanlah Inventory.java dan kumpulan filenya!

File pendukung:


- [Box.java](#)
- [Inventory.java](#)

Contoh driver:

```

```
Inventory inventory2 = new Inventory(new ArrayList<Box<?>>());
inventory2.openAll();
Box<String> temp = new Box<>("Hello, World!");
Box<Double> temp2 = new Box<>(321.32323);
inventory2.addBox(temp);
inventory2.addBox(temp2);
inventory2.openAll();
```
```

Java 8 ▾

 [Inventory.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict  | Description        |
|----|-------|----------|--------------------|
| 1  | 20    | Accepted | 0.06 sec, 27.81 MB |
| 2  | 20    | Accepted | 0.07 sec, 28.25 MB |
| 3  | 20    | Accepted | 0.06 sec, 30.80 MB |
| 4  | 20    | Accepted | 0.07 sec, 28.34 MB |
| 5  | 20    | Accepted | 0.13 sec, 28.40 MB |

## Question 3

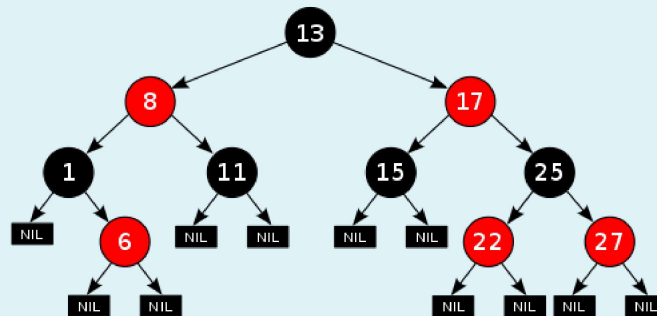
Correct

Mark 100.00 out of 100.00

|              |       |
|--------------|-------|
| Time limit   | 1 s   |
| Memory limit | 64 MB |

## How to Make Million Before Roro Dies

Bondowoso sangat ingin membahagiakan Roro dengan kekayaan sehingga kini Bondowoso merangkap sebagai kontraktor candi. Karena menjadi kontraktor, Bondowoso memiliki banyak kontak dari jin-jin nya yang sigap bekerja jika menerima pesanan pembuatan candi, dan hal ini membuat dia kewalahan dengan semakin banyak kontak yang ada. Melihat hal ini, Bondowoso berinisiatif untuk membuat sebuah kontak manajer untuk mengelola kontak jin.



Ilustrasi TreeSet untuk tipe data Integer

Bondowoso ingin agar daftar kontak yang **dibuat bersifat unik case sensitive untuk nama** dan **terurut sesuai abjad huruf**. Oleh karena itu, dia membuat daftar kontak menggunakan **TreeSet**. Tugas anda adalah **melengkapi ContactManager.java** untuk membantu Bondowoso membuat manajer kontak. Pastikan anda mengeluarkan **Exception** yang sesuai untuk tiap kasus sesuai deskripsi yang ada di ContactManager.java.

Hint: **Kecermatan membaca dokumentasi** akan sangat membantu pengerjaan soal ini. Dengan method yang tepat, anda dapat menyelesaikan soal ini dengan mudah.

Link soal:

[how\\_to\\_make\\_million\\_before\\_roro\\_dies.zip](#)

[ContactManager.java](#)

\*diberikan pula contoh file Main sehingga kalian tidak perlu membuat sendiri dan semoga mempercepat pengerjaan :D

Java 8

[ContactManager.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict  | Description        |
|----|-------|----------|--------------------|
| 1  | 10    | Accepted | 0.07 sec, 28.89 MB |
| 2  | 10    | Accepted | 0.07 sec, 28.05 MB |
| 3  | 10    | Accepted | 0.06 sec, 28.49 MB |

| No | Score | Verdict  | Description        |
|----|-------|----------|--------------------|
| 4  | 10    | Accepted | 0.07 sec, 29.06 MB |
| 5  | 10    | Accepted | 0.07 sec, 28.79 MB |
| 6  | 10    | Accepted | 0.07 sec, 28.46 MB |
| 7  | 10    | Accepted | 0.15 sec, 28.79 MB |
| 8  | 10    | Accepted | 0.08 sec, 28.18 MB |
| 9  | 10    | Accepted | 0.07 sec, 30.69 MB |
| 10 | 10    | Accepted | 0.07 sec, 29.03 MB |

Question **4**

Partially correct

Mark 60.00 out of 100.00

|              |       |
|--------------|-------|
| Time limit   | 1 s   |
| Memory limit | 64 MB |

**Menkrep Crafting**

Level: Medium

Bandung Bondowoso sedang suka bermain menkrep karena dia bisa melakukan crafting. Karena Bandung Bondowoso baru saja belajar OOP, ia mencoba membuat crafting table di Java. Coba bayangkan kamu adalah Bandung Bondowoso dan cobalah buat crafting table-mu! Kumpulkan file-file tersebut **kecuali Material.java, Stick.java, Iron.java** dalam zip yang bernama craft.zip.

Note:

Terdapat perubahan keterangan method use pada Hoe.java

[File-file pendukung.](#)

Contoh driver:

...

```
Crafting c5 = new Crafting();
c5.addMaterial(new Stick());
c5.addMaterial(new Stick());
c5.addMaterial(new Iron());
c5.addMaterial(new Iron());
c5.addMaterial(new Iron());
try {
    Axe a2 = c5.craftAxe(2, 1, 2);
    a2.attack();
} catch (Exception e) {
    System.out.println(e.getMessage());
}
try {
    Sword s2 = c5.craftSword(2, 1, 5);
    s2.attack();
} catch (Exception e) {
    System.out.println(e.getMessage());
}
c5.removeIron(1);
...
```

Java 8 ▾

 [craft.zip](#)

Score: 60

Blackbox

Score: 60

Verdict: Wrong answer

Evaluator: Exact

| No | Score | Verdict      | Description        |
|----|-------|--------------|--------------------|
| 1  | 20    | Accepted     | 0.40 sec, 26.26 MB |
| 2  | 0     | Wrong answer | 0.23 sec, 28.12 MB |
| 3  | 20    | Accepted     | 0.15 sec, 28.27 MB |

| No | Score | Verdict      | Description        |
|----|-------|--------------|--------------------|
| 4  | 20    | Accepted     | 0.14 sec, 29.80 MB |
| 5  | 0     | Wrong answer | 0.22 sec, 28.15 MB |



## Question 5

Correct

Mark 100.00 out of 100.00

|              |       |
|--------------|-------|
| Time limit   | 1 s   |
| Memory limit | 64 MB |

Diberikan kode [JsonField.java](#) sebagai berikut dan kode [JsonLoader.java](#) sebagai berikut

Upload JsonLoader.java dengan telah mengimplementasikan method `serializeObject` sesuai keterangan yang diberikan. Contoh hasil sebagai berikut

```
```java
```

```
public class Person {
    @JsonField(name = "full_name") private String name;
    @JsonField(serialized = false) private int age;
    private boolean isStudent;
    private String[] hobbies;
    private List<String> friends;
    private Map<String, String> address;

    public Person(String name, int age, boolean isStudent, String[] hobbies, List<String> friends, Map<String, String> address) {
        this.name = name;
        this.age = age;
        this.isStudent = isStudent;
        this.hobbies = hobbies;
        this.friends = friends;
        this.address = address;
    }
}
...
```
```

```
```java
```

```
Person person = new Person(
    "John Doe",
    25,
    true,
    new String[] {"Reading", "Coding"},
    Arrays.asList("Alice", "Bob", "Charlie"),
    new HashMap<String, String>() {
        {
            put("city", "New York");
            put("state", "NY");
        }
    }
);

JsonLoader jsonLoader = new JsonLoader();
String json = jsonLoader.serialize(person);
// {"full_name":"John Doe","isStudent":true,"hobbies":["Reading","Coding"],"friends":["Alice","Bob","Charlie"],"address":{"city":"New York","state":"NY"}}
...
```
```

Java 8

 [JsonLoader.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

| No | Score | Verdict  | Description        |
|----|-------|----------|--------------------|
| 1  | 9     | Accepted | 0.09 sec, 29.31 MB |
| 2  | 9     | Accepted | 0.06 sec, 28.02 MB |
| 3  | 9     | Accepted | 0.06 sec, 30.87 MB |
| 4  | 9     | Accepted | 0.06 sec, 30.79 MB |
| 5  | 9     | Accepted | 0.07 sec, 28.44 MB |
| 6  | 9     | Accepted | 0.07 sec, 30.96 MB |
| 7  | 9     | Accepted | 0.07 sec, 29.06 MB |
| 8  | 9     | Accepted | 0.09 sec, 30.49 MB |
| 9  | 9     | Accepted | 0.11 sec, 28.21 MB |
| 10 | 9     | Accepted | 0.08 sec, 28.45 MB |
| 11 | 10    | Accepted | 0.08 sec, 28.36 MB |

|              |       |
|--------------|-------|
| Time limit   | 1 s   |
| Memory limit | 64 MB |

**Trick or Thread!**

Bondowoso memiliki 240 buah kotak yang berisi kado rahasia yang terdapat sebuah hadiah utama pada 1 kotaknya dan hadiah biasa pada 239 kotak sisanya. Karena anda sudah menyelesaikan tugas besar dengan baik, anda diperkenankan untuk membuka kado rahasia tersebut. Namun, tentu nafsu manusiawi jika kita menginginkan hadiah utama dibandingkan hadiah biasa sehingga anda meminta ijin ke Bondowoso untuk membiarkan anda mencari hadiah utama. Bondowoso menyanggupi dengan syarat untuk melakukan pencarian hadiah utama, anda harus menggunakan program Bondowoso.

Program Bondowoso akan menerima sebuah angka antara 1 hingga 240, kemudian akan memberikan respon **true jika hadiah utama**, dan **false jika hadiah biasa**. Namun, karena program Bondowoso masih berjalan di Windows XP, program Bondowoso mengalami delay. Anda, yang seorang mahasiswa Informatika, memiliki ide untuk membuat sebuah multithreading dari program Bondowoso sehingga bisa mencari hadiah utama lebih cepat. Namun, karena komputer Bondowoso masih berupa Windows XP, komputer hanya **maksimum menjalankan 8 Thread**.

Diberikan beberapa class, tugas anda adalah **memodifikasi class MultithreadProgram.java**. Anda juga boleh memodifikasi atribut serta method yang ada (seperti menambahkan abstract, synchronized, dsb). Jika tidak dimodifikasi, pada kasus worst case program bisa mencapai 2.4 detik. Namun, jika dimodifikasi dengan benar, program bisa mencapai dibawah 1 detik. Waktu di Olympia akan di set 1 detik sehingga jika tidak dimodifikasi, **test case akan mengalami TLE (Time Limit Exceeded)**.

Hint: keyword **volatile** dalam bahasa Java digunakan untuk menunjukkan bahwa nilai suatu variabel dapat diubah oleh beberapa thread secara bersamaan

Link soal:

[trick or thread.zip](#)

[MultithreadProgram.java](#)

Contoh file main:

```
public class Main {
    public static void main(String[] args) {
        GiftChecker checker = new GiftChecker(1, 10);
        MultithreadProgram mtChecker = new MultithreadProgram(checker);
        mtChecker.findCorrectNumber();
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
        System.out.println("Number found: " + mtChecker.foundNumber);
    }
}
```

Java 8 ▾

 [MultithreadProgram.java](#)

Score: 0

Blackbox

Score: 0

Verdict: Compile error

Evaluator: Exact

| No | Score | Verdict       | Description   |
|----|-------|---------------|---|
|    | 0     | Compile error | javac Main.java Computer.java GiftChecker.java MultithreadProgram.java Main.java:56: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:68: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:80: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:92: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:104: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:116: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:128: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:140: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:152: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>Main.java:164: error: unreported exception InterruptedException; must be caught or declared to be thrown mtChecker.findCorrectNumber(); ^<br>10 errors<br>Makefile:27: recipe for target 'Main.class' failed make: *** [Main.class] Error 1 |

[← Latihan Praktikum 6 & 7](#)

Jump to...

